CHAPTER VI

MAX COVER PERIOD ASSIGNMENT- LOT SIZE HEURISTIC (MCP-LS)

The aim of this chapter is to provide an alternated heuristic for the research problem. Previously, A-LS and PA-LS were presented and discussed. The drawbacks of those heuristics are no improvement after early iterations in the A-LS heuristic and too long solving time in the PA-LS heuristic. Understand by this, the challenge is to develop heuristic that give a better satisfactory solution than A-LS heuristic while using lesser solving time than PA-LS heuristic. Notice that PA-LS heuristic is an improvement version of A-LS in the aspect of solution performance. Then, improvement of PA-LS heuristic in the way of reduction solving time is the goal of this chapter.

This chapter is organized as follows: The introduction is presented in section 6.1. The heuristics are described in Section 6.2. In Section 6.3, we present the heuristics procedures that were used. The computational results are given in Section 6.4. Finally, some concluding remarks are provided in Section 6.5.

6.1 Introduction

The goal of this chapter is to find the way to improve the PA-LS heuristic. On basis concept of PA-LS heuristic, the heuristic method changes the Lot-for-Lot solution to the solution with consideration holding by relaxing the assignment matrix, on believing that the closer to original assignment will give the better solution. As the assignment matrix is clearly the key to change the solving time, the new way of relaxing assignment matrix which is a max cover period assignment will be discussed and developed in this chapter. The proposed heuristic method in this is called Max

Cover Period Assignment – Lot size (MCP-LS). The computational test is used to analyze its solution performance and solving time performance.

6.2 Heuristic Description

The PA-LS heuristic is applied the assignment matrix relaxation to the original assignment matrix as presented in chapter 5. It's result in long solving time. Back to the basic concept about possible lot size as presented in chapter 4, lot size can possible in the range from zero to the minimum value between capacity and summation of next demands. It indicates that the lot size can be changed by changing assignment matrix whenever there are possible available in the range as mentioned. Likewise, the available capacity should be the strong boundary of the lot size in each period. Understanding by this, the assignment matrix should be changed whenever it has a potential to change the lot size and consequently solution of the problem. As far as the quality of the approximation is concerned, the max cover period calculation is proposed to finding the potential changed assignment matrix.

6.3 Heuristic Procedures

The aim of this section is to describe Max Cover Period – Lot size heuristic on believing that it uses lower running time. Based on PA-LS heuristic, the steps of PA-LS consist of the partial assignment on given lot size phase and the lot size on given assignment phase. As the lot size phase uses the assignment matrix as input data, there is a need to find what assignment matrix should be. The Max Cover Period calculation is used to find the relax assignment matrix. The flow of this heuristic is shown in Figure 6.1.

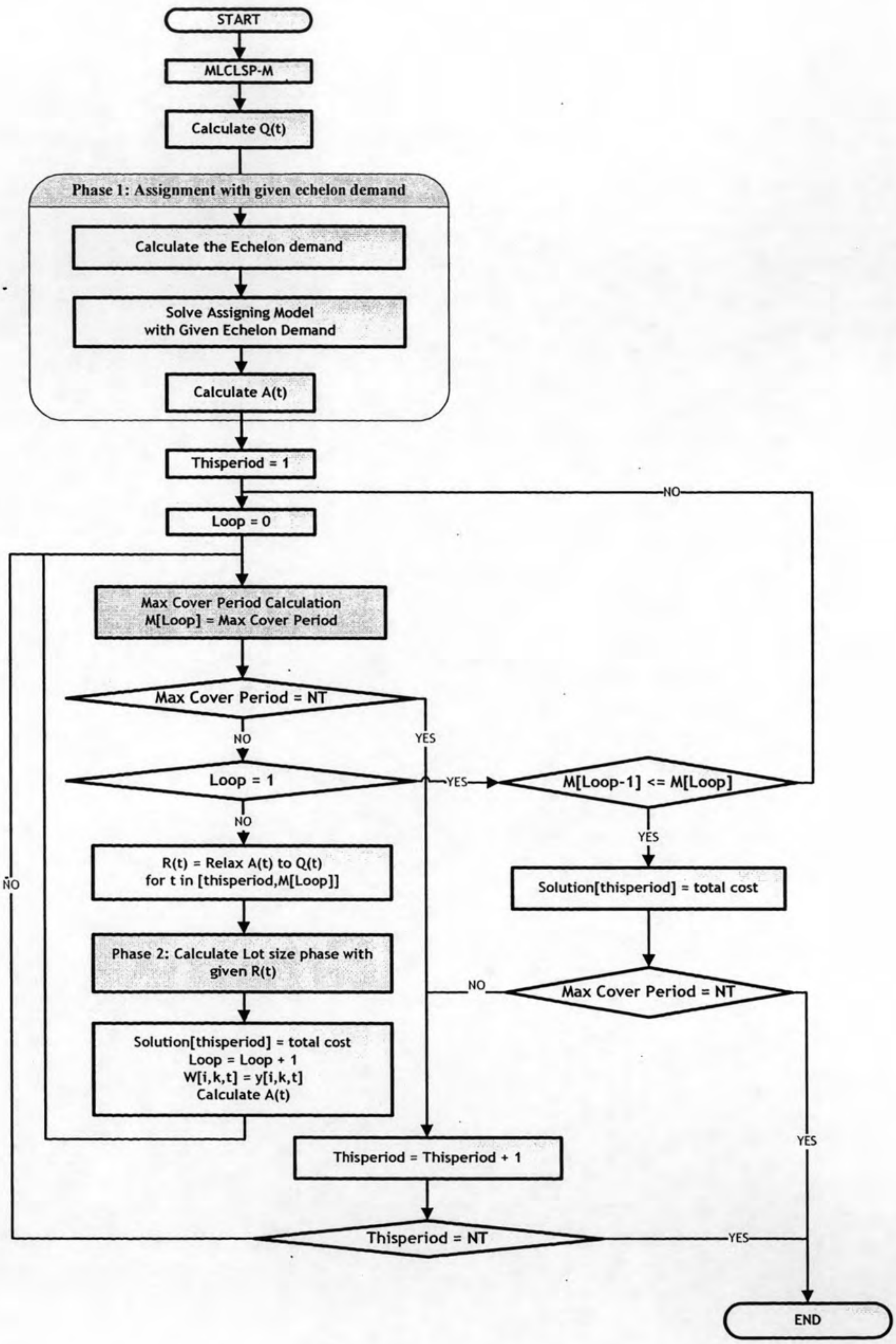

Figure 6.1 Illustration of MCP-LS heuristic flow

6.3.1 Phase 1: Echelon Assignment problem with given lot-size

With the same problem description, assumption and assignment model as the PA-LS heuristic presented in section 5.3.1, the result of this part is also the assigning matrix ($A(t)$) or the pair of item-workstation to be used as input data in the lot sizing part. All concepts in this phase are based on PA-LS except that the max cover period is implemented. The max cover period will be described in detailed later. This phase will be done only in the first iteration. The given assignment matrix for other iterations automatically generated in the Lot size with given assignment phase.

6.3.2 Max Cover Period Calculation

Between the two phases, Max Cover Period Calculation is introduced to find the Relax assignment matrix ($R(t)$). The flow of this calculation is presented as shown in Figure 6.2. The steps of this phase are presented as follows:

Step 1: Calculate Cover period for item-workstation

Let $thisperiod$ represent the considered period, $Availcap_{thisperiod}^{(i,k)}$ represents the available capacity of workstation k for operating item i in the considered period or $(i,k) \in Q(t)$. The $Availcap_{thisperiod}^{(i,k)}$ can be calculated by equation (C) as follows:

$$Availcap_{thisperiod}^{(i,k)} = \begin{cases} cap_{thisperiod}^{k} - \sum_{t=thisperiod}^{Coverperiod_{thisperiod}^{(i,k)}} \left(g_t^{i,k} + o_t^{i,k} D_t^i\right), \\ \quad \forall (i,k) \in Q(thisperiod), \forall k \in WP \\ cap_{thisperiod}^{k} - \sum_{t=thisperiod}^{Coverperiod_{thisperiod}^{(i,k)}} \left(D_t^i\right), \\ \quad \forall (i,k) \in Q(thisperiod), \forall k \in WR \end{cases} \quad \text{(C)}$$

The cover period can be found by checking the available capacity of workstation ($Availcap_{thisperiod}^{(i,k)}$). By the equation, the cover period ($Coverperiod_{thisperiod}^{(i,k)}$) represents the minimum period that makes $Availcap_{thisperiod}^{(i,k)} \le 0$.

Step 2: Calculate Max Cover Period.

Obviously, the cover period for each workstation can be varied by the item assigned. The algorithm is need only one value to represent the cover period of each workstation, so we use the max value of the cover period for all items assigned on the workstation. Let $MaxCoverPeriod^{k}_{thisperiod}$ represents Max cover period of workstation k in this considered period. Max cover period can be found by equation (M)

$$MaxCoverPeriod^{k}_{thisperiod} = \underset{(i,k)\in Q(t)}{Max}\left(Coverperiod^{(i,k)}_{thisperiod}\right) \qquad \text{(M)}$$

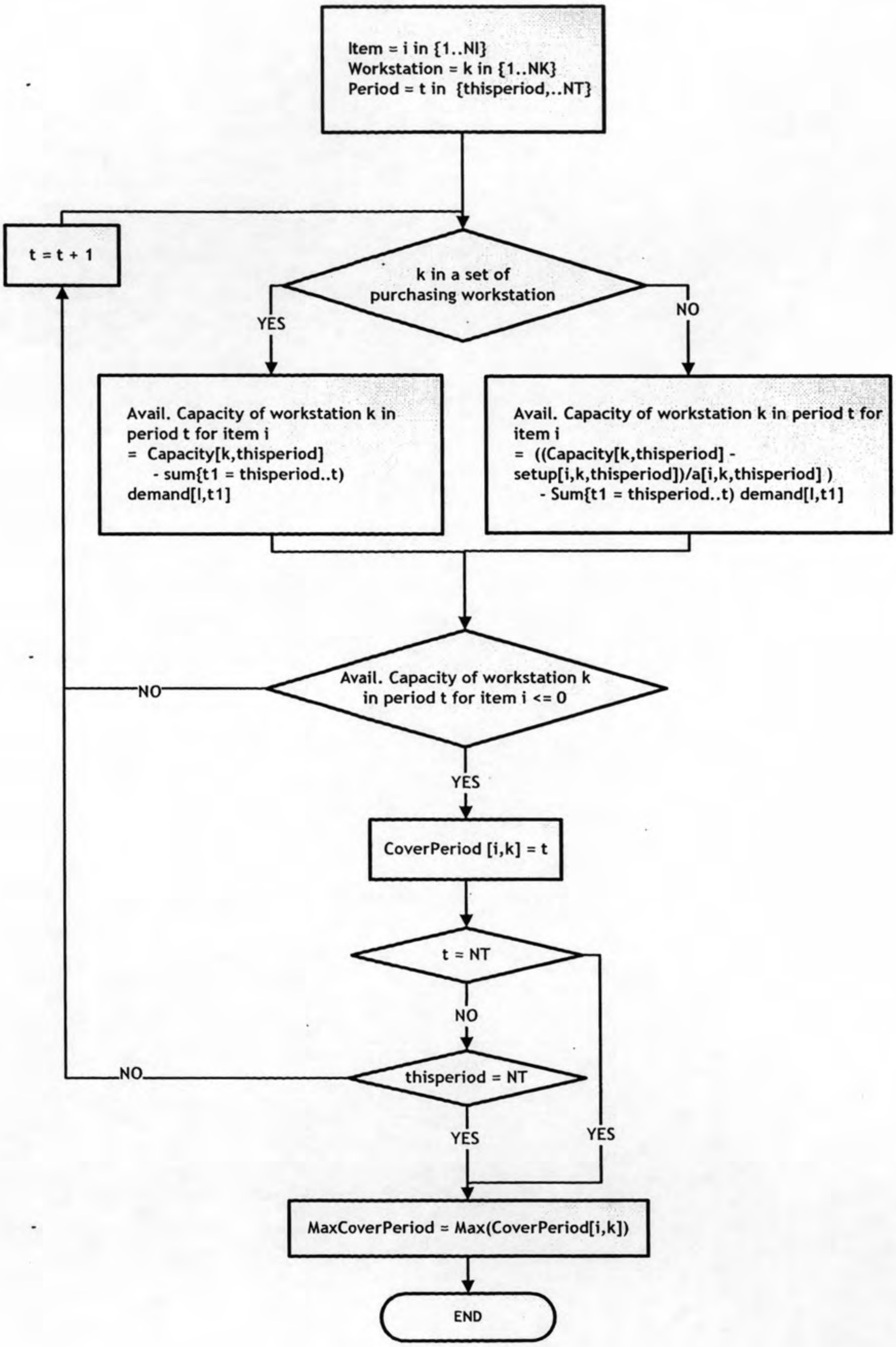

Figure 6.2 Illustration of Max Cover Period Calculation

6.3.3 Phase 2: Lot sizing problem with given Max Cover period assignment matrix

The lot size model is the same as A-LS and PA-LS. The heuristic differs from PA-LS only in the part of relaxation assignment matrix. The cover period technique is introduced to find the number of cover periods that the assignment matrix ($A(t)$) should be replaced by the original matrix ($Q(t)$). It is based on the concept of production may satisfy one period demand or more than one period until the limited capacity. The capacity of each item-workstation assignment will tell us the number of periods it can cover demands. If the assignment is changed more than the cover period, it has no effect on the solution and wastes the solving time for nothing. Likewise, the assignment matrix should be changed to the original matrix, only whenever the pairs of item-workstation have a potential to give a better solution. Therefore, let set $R(t)$ is the Relaxation Assignment Matrix and it can be defined as follows:

$$R(t) = \begin{cases} Q(t) & \forall t \in \{thisperiod,..,MaxCoverPeriod^{k}_{thisperiod}\} \\ A(t) & otherwise \end{cases} \quad \text{(R)}$$

In this phase, lot size model will be solved with given $R(t)$ and the solution will keep as this considered period solution until we can find the better solution or meet the condition in loops and iterations. The detailed of loops and iterations is presented in the next section.

6.3.4 Loops and iterations

To make sure that $MaxCoverPeriod^{k}_{thisperiod}$ is valid for this assignment; we calculate it in a loop. In the first time, we set $Loop = 0$ and calculate $MaxCoverPeriod^{k}_{thisperiod}$ as above step and keep the solution as $M[Loop]$. Next we solve the lot size model with the Relaxation assignment matrix ($R(t)$). Set $Loop = Loop + 1$ and let $w^{i,k}_{t} = y^{i,k}_{t}$. Then, we calculate set $A(t) = \{(i,k) \mid w^{i,k}_{t} = 1\}$, which is a set of the assignment item-workstation in period. We will run this considered period again and also let $M[Loop] = MaxCoverPeriod^{k}_{thisperiod}$. If $M[Loop-1] < M[Loop]$ then set $Loop = 0$ and run the algorithm again, otherwise we go to next period consideration.

6.3.5 Termination

The termination of MCP-LS heuristics occurs in two ways. First, all periods are considered or the value of $MaxCoverPeriod^{k}_{thisperiod}$ equal the last period of planning horizon. In these iterations, we use the AMPL/CPLEX 8.0.0 to solve. Second, it's run until meet the user iteration limited.

6.4 Computational Experiment

In this section, we first analyze the quality of the solutions obtained by the heuristic method described in Section 6.3. We then compare the results from the approaches method. Finally we consider computing times.

We consider a scenario and use the test instance as same as in section 4.4. To show the computational test of this heuristic we also follow the step as presented in section 4.4. First we select on problem to show the improvement between iterations as shown in Figure 6.3. Second, we compare the solution from solving MLCLSP-M and MCP-LS in three aspects which are solving time, total cost and %different of total cost between the heuristics solution and mathematical model solution. The solving time and total cost aspects are shown in Figure 6.4 and Figure 6.5. Because %different of total cost between MCP-LS heuristics solution and mathematical model solution is equally for all test instances, we present it in Table 6.1 instead of in graph format. Table 6.1 also presents the %different of solving time between MCP-LS heuristics solution and mathematical model solution.

We use problem size 10-item 10-workstation with eight periods (the product structure is same as the standard library) to show this problem. To solve this Mixed Integer Programming (MIP) problem, we used AMPL/CPLEX 8.0.0 solver. Figure 6.3 shows example of one problem instance AG01130 showing improvement between iterations. Second, we compare the solution from solving MLCLSP-M and A-LS in three aspects which are %different of total cost, %different of running time and

%running time between the heuristics solution and mathematical model solution. The results are classified by the capacity profiles of the problem instance.

It can be observed from Figure 6.3 for the problem instance AG01130 example that solution is improved dramatically in the early iterations and then stay constantly until end. The comparison between MLCLSP-M and A-LS is shown in Table 6.1.

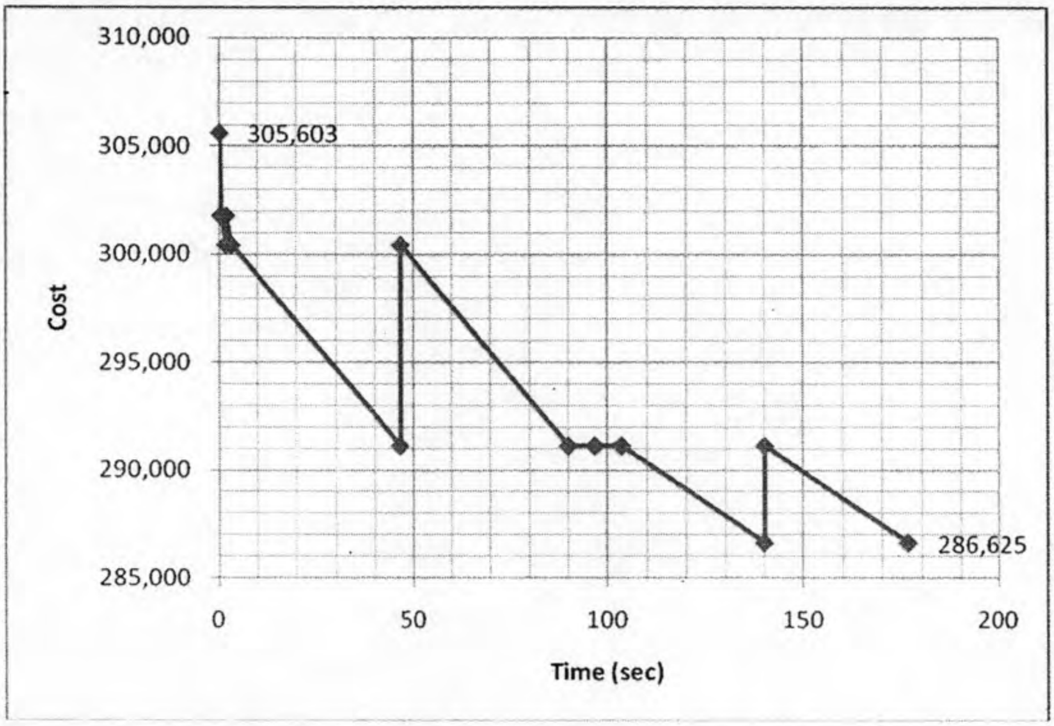

Figure 6.3 The improvement between iterations sample for instance AG01130

Table 6.1 The comparison between MLCLSP-M and MCP-LS for problem AG01130

	MLCLSP-M	MCP-LS Heuristic	%diff.
Cost($)	286,623	286,625	0.0007%
Time(sec.)	8,232.16	176.6563	98%

The capacity profiles of the test instances are constant utilization, varying utilization by product structure level or BOM, and varying by periods. Comparison of results for these capacity profiles between MLCLSP-M and MCP-LS are presented in

Table 6.2, Table 6.3 and Table 6.4, respectively. For more clearly, we present the summary of MCP-LS for different capacity profiles in Figure 6.3.

As seen in Table 6.2, Table 6.3 and Table 6.4, the solutions from MLCLSP-M and MCP-LS are much closed. The maximum different of total cost between MLCLSP-M and MCP-LS is 1.428%. The average different is 0.075%, while minimum different is only 0%. The results also show that the MCP-LS heuristic results with low capacity utilization profiles using solving time and performance of the solution almost or equally with the MLCLSP-M. Understanding by this, in that case, the max cover period is equal to the planning horizon of the problems. Therefore, the problem profiles have an effect on solving time and on performance of the solution. To sum up, the heuristic give almost optimal solution (average 0.075% different) within less solving time on average (38% different).

Table 6.2 The comparison between different capacity profiles with given constant utilization capacity profiles

Capacity Profile	%Diff Cost			%Diff Time			% Running Time		
	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.	Max
Constant 90%	0.00%	0.37%	1.43%	57.50 %	83.07 %	99.52 %	0.48%	16.93 %	42.50 %
Constant 70%	0.00%	0.02%	0.07%	48.16 %	79.43 %	99.30 %	0.70%	20.57 %	51.84 %
Constant 50%	0.00%	0.00%	0.00%	0.00%	18.88 %	65.99 %	34.01 %	81.12 %	100.00 %

Table 6.3 The comparison between different capacity profiles with given varying by level capacity profiles

Capacity Profile	%Diff Cost			%Diff Time			% Running Time		
	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.	Max
Vary 90/70/50 by level	0.00%	0.00%	0.00%	0.00%	5.50%	16.81 %	83.19 %	94.50 %	100.00 %
Vary 50/70/90 by level	0.00%	0.02%	0.10%	0.00%	22.36 %	97.18 %	2.82%	77.64 %	100.00 %

Table 6.4 The comparison between different capacity profiles with given varying by period capacity profiles

Capacity Profile	%Diff Cost			%Diff Time			% Running Time		
	Min	Avg.	Max	Min	Avg.	Max	Min	Avg.	Max
Increasing by period	0.00%	0.12%	0.51%	0.00%	15.84 %	34.85 %	65.15 %	84.16 %	100.00 %
Decreasing by period	0.00%	0.00%	0.01%	0.00%	15.42 %	77.09 %	22.91 %	84.58 %	100.00 %

Table 6.5 The average solving time comparison between MLCLSP-M and A-LS for all problems

	MLCLSP-M	MCP-LS Heuristic	%diff.
Average Time(sec.)	10,239.09 (2.80 hr.)	6335.92	38%

In Table 6.5, the cost from the heuristic is slightly different from the MLCLSP-M but uses dramatically lesser computational time in all problems. The solving time comparison Illustration of summary of MCP-LS for different capacity profiles is shown in Figure 6.4.

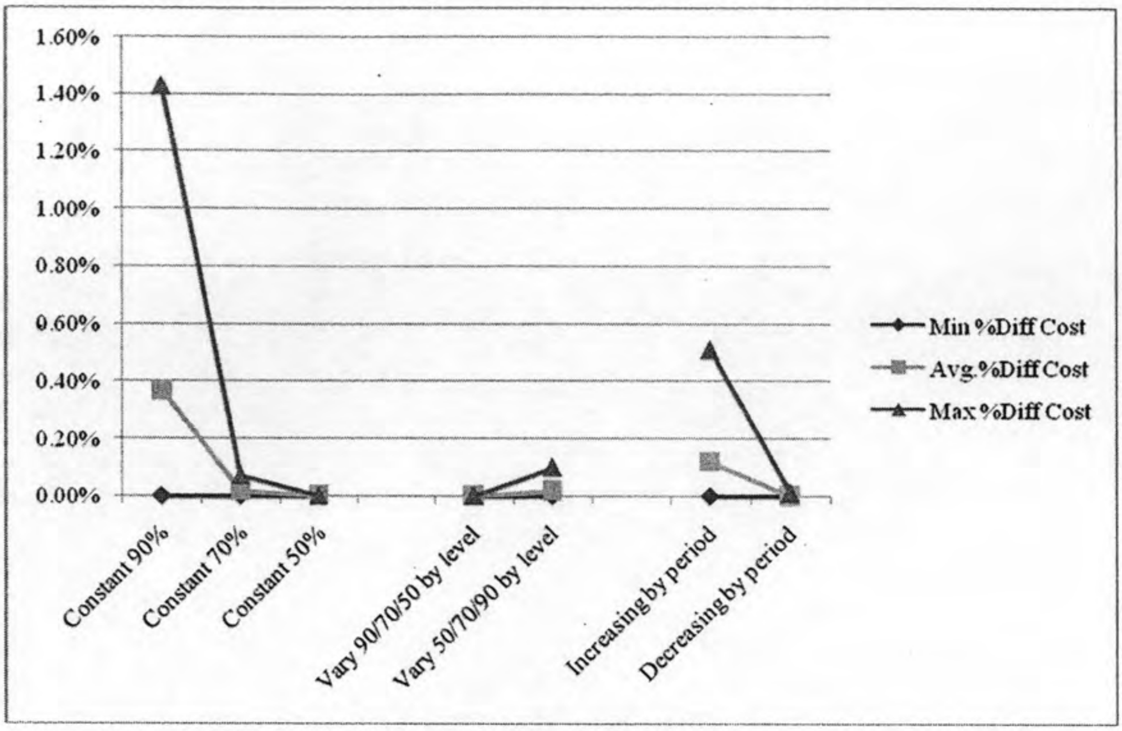

Figure 6.4 Illustration of summary % different cost of MCP-LS for different capacity profiles

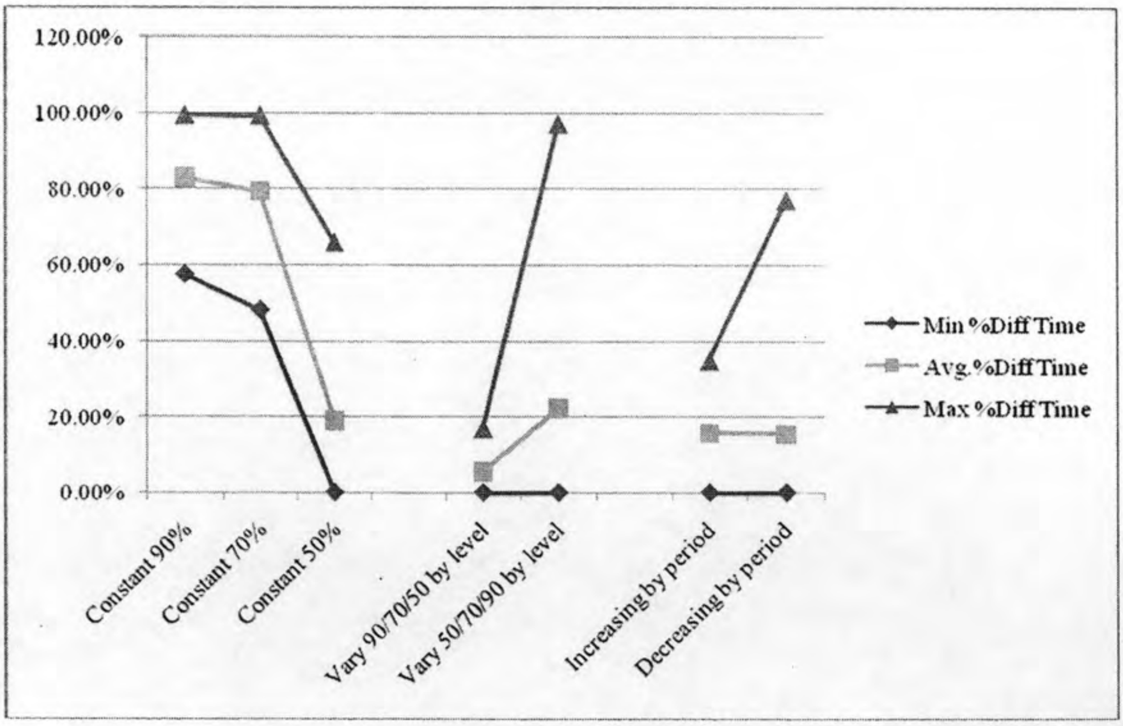

Figure 6.5 Illustration of summary %different time of MCP-LS for different capacity profiles

As seen in Figure 6.4 and Figure 6.5, although over all the solution performance is good with maximum 1.43% different cost form optimal solution, the running time is high for the low utilization capacity profiles.

6.5 Conclusion

MCP-LS heuristic is developed and analyzed in this chapter. It's obvious that the solution is very close to the optimal solution and the time for solving is lesser than PA-LS. The drawbacks of this heuristic is that it uses longer time in the case of low utilization workstation profile. With low improvement of solution in the later iteration, this drawback can be fixed by set limited iteration. This heuristic is the best of all proposed heuristics with running time and solution performance criteria.